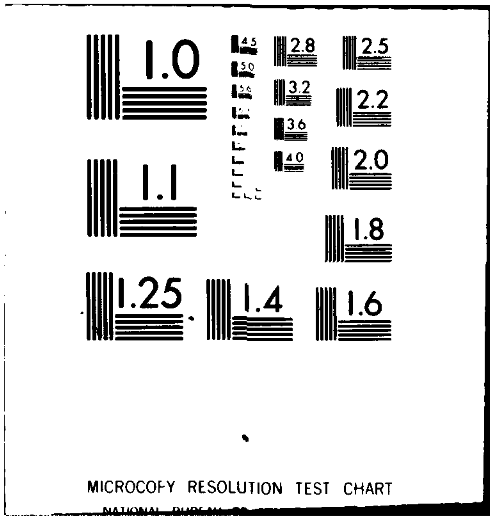
AD-A087 641 SRI INTERNATIONAL MENLO PARK CA COMPUTER SCIENCE LAB F/G 9/2
A CATALOGUE OF CANONICAL TERM REWRITING SYSTEMS.(U)
APR 80 J HULLOT F49620-79-C-0099
UNCLASSIFIED SRI/CSL-113 AFOSR-TR-80-0460 NL

1 of 1
AD A087641

END
DATE
FILMED
9-80
DTIC



MICROCOFY RESOLUTION TEST CHART

NATIONAL BUREAU OF

A CATALOGUE OF CANONICAL TERM REWRITING SYSTEMS

Technical Report CSL-113

April 1980

By: Jean-Marie Hullot, Institute Observer
Computer Science Laboratory
Computer Science and Technology Division

SRI International
333 Ravenswood Avenue
Menlo Park, California 94025
(415) 326-6200
Cable: SRI INTL MPK
TWX: 910-373-1246

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFOSR-TR- 80-0460	2. GOVT ACCESSION NO. AD-A087641
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) A CATALOGUE OF CANONICAL TERM REWRITING SYSTEMS	5. TYPE OF REPORT & PERIOD COVERED Interim
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jean-Marie Hullot	8. CONTRACT OR GRANT NUMBER(s) F49620-79-C-0099
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International 333 Ravenswood Avenue Menlo Park, CA 94025	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A2
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research/NM Bolling AFB, Washington, DC 20332	12. REPORT DATE April 1980
	13. NUMBER OF PAGES 35
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

IT IS ATTEMPTED

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

~~We attem~~pt in this paper to put together some experimental results obtained by Knuth and Bendix, Lankford and Ballantyne, Peterson and Stickel, as well as to present some new results. ~~We~~ IT does not give a theoretical study of the methods used. Thus the reader is assumed to be familiar with the original Knuth and Bendix algorithm and it extnesions.

UNCLASSIFIED

SRI International

6

A CATALOGUE OF CANONICAL TERM REWRITING SYSTEMS,

ACCESSION for		
NTIS	White Section	☒
DDC	Buff Section	□
UNANNOUNCED		□
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or	SPECIAL
A		

14 SRI/CSL-113

9 Technical Report CSL-113

11 April 1980 36

10

By: Jean-Marie Hullot, Institute Observer
Computer Science Laboratory
Computer Science and Technology Division

18 AFOSR 19 TR-80-0460

This research was supported by the Air Force Office of Scientific Research under Contract F49620-79-C-0099

15 16 2304 17 A2

410646

A CATALOGUE
OF
CANONICAL TERM REWRITING SYSTEMS

Jean-Marie Hullot
INRIA and SRI International

Contents

0. Introduction.

We attempt in this paper to put together some experimental results obtained by Knuth and Bendix (70), Lankford and Ballantyne (77a, 77b, 77c), Peterson and Stickel (77), as well as to present some new results. We do not give a theoretical study of the methods used. Thus the reader is assumed to be familiar with the original Knuth and Bendix algorithm and its extensions. Descriptions and justifications of these algorithms may be found in the previous references and in the following: Huet (77), Huet and Oppen (80) and Hullot (80). Note that another extension of the Knuth and Bendix algorithm is given in Huet (77); however we do not study practical applications of this method here. Note also that we do not study any examples of "permutative reductions" other than associative and commutative reductions in which case we use the method defined by Peterson and Stickel (77). Finally, one can find a proof of corrections of all these algorithms in Huet (80).

The program used to run these examples is written in VLISP, a dialect of the LISP programming language developed at Université de Vincennes by Patrick Greussay and Jérôme Chailloux. It includes Knuth and Bendix's algorithm and Peterson and Stickel's algorithm in case of commutativity and associativity. Examples 6 to 19 require the use of the associative and commutative case. This program is running interpreted on a DEC KL 10.

1. Groupoids.

A *groupoid* is a set with a single binary operation denoted by $\cdot$. As an example of a groupoid with special properties, we shall study the case of *central groupoids*, i.e. groupoids satisfying the following axiom:

$$(x \cdot y) \cdot (y \cdot x) = y. \qquad (a1)$$

Using the completion algorithm, a canonical term rewriting system is obtained for central groupoids (see Knuth and Bendix (70) examples 6 and 16):

$$(x \cdot y) \cdot (y \cdot x) \rightarrow y; \qquad (r1)$$

$$x \cdot ((x \cdot y) \cdot z) \rightarrow x \cdot y; \qquad (r2)$$

$$(x \cdot (y \cdot z)) \cdot z \rightarrow y \cdot z. \qquad (r3)$$

Note that the two new rules have been derived by superposing the first one on itself.

2. Quasigroups and Loops.

Before introducing the associativity law, we study the theory of quasigroup. We have three binary operators $\cdot$, $\backslash$ and $/$ satisfying the following equations:

$$x \cdot (x \backslash y) = y; \qquad (a1)$$

$$(x \,/\, y) \cdot y = x; \qquad (a2)$$

$$x \backslash (x \cdot y) = y; \qquad (a3)$$

$$(x \cdot y) \,/\, y = x. \qquad (a4)$$

Classically, $(a1)$ is interpreted as: *for all x and y, there exists $z = x \setminus y$ such that $x \cdot z = y$.* $(a3)$ allows us to say that there exists *at most one such z*. Indeed assume there exists two such elements, say z_1 and z_2 then:

$$z_1 = x \setminus (x \cdot z_1) = x \setminus (x \cdot z_2) = z_2.$$

Obviously axioms $(a2)$ and $(a4)$ are the left-right dual of $(a1)$ and $(a3)$.
These four equations are proposed to the program as left to right rewrite rules $(r1)$ to $(r4)$. A new rule is then derived from $(r3)$ and $(r2)$:

$$(x \,/\, y) \setminus x \to y, \tag{r5}$$

and its dual from $(r4)$ and $(r1)$:

$$x \,/\, (y \setminus x) \to y. \tag{r6}$$

This set of six rules is a canonical term rewriting system for quasigroup theory. We now use this canonical term rewriting system to study some classes of quasigroups with identity. As a first example, we study abstract loops (see Evans (51) and Knuth and Bendix (70)), i.e. quasigroup with identity. We present to the program the two new rewrite rules:

$$e \cdot x \to x; \tag{r7}$$

$$x \cdot e \to x. \tag{r8}$$

Four new rules are immediately found, leading to a canonical term rewriting system for abstract loops:

$$e \setminus x \to x; \tag{r9}$$

$$x \,/\, e \to x; \tag{r10}$$

$$x \,/\, x \to e; \tag{r11}$$

$$x \setminus x \to e. \tag{r12}$$

We study now some classes of quasigroups with identity where the word problem is known to be solvable. The existence of a canonical term rewriting system is another proof of this result.
We begin with the idempotency law given as the following rewrite rule:

$$x \cdot x \to x. \tag{r7}$$

This set of seven rules is completed in a canonical term rewriting system with the two following rules:

$$x \setminus x \to x; \tag{r8}$$

$$x \,/\, x \to x. \tag{r9}$$

Note that one could obtain the same set in presenting ($r8$) or ($r9$) instead or ($r7$). That is, in a quasigroup the idempotency law for one of the three binary operators implies the idempotency for the other two.
Instead of idempotency, we give now unipotency for $\cdot$, that is:

$$x \cdot x \to 1, \tag{r7}$$

where 1 is a new constant symbol. Once more the set is completed with two rules:

$$x \setminus 1 \to x; \tag{r8}$$

$$1 \,/\, x \to x. \tag{r9}$$

Note that this time the three binary operators do not play symmetrical roles for the new identities.
As a last example for quasigroup theory, we introduce the semisymmetry law for $\cdot$, that is:

$$x \cdot (y \cdot x) = y. \tag{r7}$$

The first equation derived by the program is the following:

$$(x \setminus y) \cdot y \to x. \tag{r8}$$

Using the interpretation given at the beginning of this section one can deduce from this identity the new identity $x \,/\, y = x \setminus y$. However, the program finds another powerful identity from ($r2$) and ($r7$):

$$x \,/\, y \to y \cdot x, \tag{r9}$$

and symmetrically from ($r3$) and ($r7$):

$$x \setminus y \to y \cdot x, \tag{r10}$$

which shows the expected property. At this step, many rules are deleted and the following set of rewrite rules appears to be canonical:

$$x \cdot (y \cdot x) \to y; \tag{r7}$$

$$x \,/\, y \to y \cdot x; \tag{r9}$$

$$x \setminus y \to y \cdot x; \tag{r10}$$

$$(x \cdot y) \cdot x \to y. \tag{r11}$$

Note that the termination of all term rewriting sytems studied in this section is obvious.

3. Semigroups and Monoids.

When the operation of a groupoid is associative, we obtain the subclass of *semigroups*. Thus, semigroup theory may be equationally defined by the following axiom:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z). \tag{a1}$$

Any orientation of this axiom as a rewrite rule leads to a canonical term rewriting system reduced to this single rule. One can now define *monoid* theory by adding to the previous theory, the two following axioms for the unit noted 1:

$$x \cdot 1 = x; \qquad (a2)$$

$$1 \cdot x = x. \qquad (a3)$$

Choosing left to right orientation for these two rules and any orientation for the associativity law, we obtain a canonical term rewriting system for free monoids.
Let us study the case of idempotent semigroups. We propose to the program the two following rewrite rules:

$$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z); \qquad (r1)$$

$$x \cdot x \rightarrow x. \qquad (r2)$$

We list below the first rules derived:

$$x \cdot (x \cdot y) \rightarrow x \cdot y; \qquad (r3)$$

$$x \cdot (y \cdot (x \cdot y)) \rightarrow x \cdot y; \qquad (r4)$$

$$x \cdot (y \cdot (x \cdot (y \cdot z))) \rightarrow x \cdot (y \cdot z); \qquad (r5)$$

$$x \cdot (y \cdot (z \cdot (x \cdot (y \cdot z)))) \rightarrow x \cdot (y \cdot z). \qquad (r6)$$

It is easy to see that the process would never terminate. We can remark on this example that our way of dealing with associativity is not general enough. However since there does not exist a canonical and *finite* associative unification algorithm we cannot hope to deal with associativity as we can do with associativity and commutativity together with the Stickel and Peterson's extension of Knuth and Bendix's algorithm.

4. Groups.

Let us define *group* theory by the following set of equations:

$$1 \cdot x = x; \qquad (a1)$$

$$x^{-1} \cdot x = 1; \qquad (a2)$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z). \qquad (a3)$$

Following Knuth and Bendix we use the completion algorithm to process this set of equations. The Knuth and Bendix ordering is used for checking termination. We give a canonical run of this example, which is slightly different from Knuth and Bendix's.
First a left to right orientation is chosen for the three given axioms, leading to rules (r1) to (r3). Then a new rule is added resulting from the superposition of rules (r3) and (r2):

$$x^{-1} \cdot (x \cdot y) \rightarrow y. \qquad (r4)$$

($r4$) is then used together with rule ($r1$) to derive:

$$1^{-1} \cdot x \rightarrow x, \qquad (r5)$$

then with rule ($r2$):

$$(x^{-1})^{-1} \cdot 1 \rightarrow x, \qquad (r6)$$

and then with rule ($r5$):

$$(1^{-1})^{-1} \cdot x \rightarrow x. \qquad (r7)$$

Superposing ($r7$) on ($r2$):

$$1^{-1} \rightarrow 1, \qquad (r8)$$

is added and this rule is used to delete rules ($r5$) and ($r7$). Two consequences of rule ($r4$) respectively with ($r6$) and itself are then found:

$$((x^{-1})^{-1})^{-1} \cdot x \rightarrow 1; \qquad (r9)$$

$$(x^{-1})^{-1} \cdot y \rightarrow x \cdot y. \qquad (r10)$$

At this step ($r6$) is replaced by:

$$x \cdot 1 \rightarrow x, \qquad (r11)$$

and ($r9$) is deleted. Note that the program has proved that 1 is a right identity. Superposing now ($r10$) on ($r11$):

$$(x^{-1})^{-1} \rightarrow x, \qquad (r12)$$

is added and used to delete ($r10$). The right inverse law is then found from ($r12$) and ($r2$):

$$x \cdot x^{-1} \rightarrow 1. \qquad (r13)$$

Then, superposing ($r12$) on ($r4$) we obtain:

$$x \cdot (x^{-1} \cdot y) \rightarrow y. \qquad (r14)$$

Note that one can obtain ($r14$) in superposing the associativity law ($r3$) on ($r13$) (as ($r4$) was obtained in superposing ($r3$) on ($r2$)). Another superposition of ($r3$) on ($r13$) gives:

$$x \cdot (y \cdot (x \cdot y)^{-1}) \rightarrow 1. \qquad (r15)$$

Then from this equation and ($r4$):

$$x \cdot (y \cdot x)^{-1} \rightarrow y^{-1}. \qquad (r16)$$

Finally, ($r15$) is deleted and the program derives from ($r4$) and ($r16$):

$$(x \cdot y)^{-1} \rightarrow y^{-1} \cdot x^{-1}. \qquad (r17)$$

This rule is used to delete ($r16$) and the set of ten remaining rules is a canonical term rewriting system for group theory. We list it below after renaming:

$$1 \cdot x \rightarrow x; \tag{$r1$}$$
$$x^{-1} \cdot x \rightarrow 1; \tag{$r2$}$$
$$(x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z); \tag{$r3$}$$
$$x^{-1} \cdot (x \cdot y) \rightarrow y; \tag{$r4$}$$
$$x \cdot 1 \rightarrow x; \tag{$r5$}$$
$$1^{-1} \rightarrow 1; \tag{$r6$}$$
$$(x^{-1})^{-1} \rightarrow x; \tag{$r7$}$$
$$x \cdot x^{-1} \rightarrow 1; \tag{$r8$}$$
$$x \cdot (x^{-1} \cdot y) \rightarrow y; \tag{$r9$}$$
$$(x \cdot y)^{-1} \rightarrow y^{-1} \cdot x^{-1}. \tag{$r10$}$$

The computer took 4.8 seconds for this calculation. Of the 14 rules derived during the process only rule ($r9$) was never used in the derivation of the canonical set. We obtain an "efficiency rating" of 92%.

Knuth and Bendix (70) study other axiomatizations of group theory.

5. (l, r) Systems.

Another interesting example studied by Knuth and Bendix is the following equational theory:

$$1 \cdot x = x; \tag{$a1$}$$
$$x \cdot x^{-1} = 1; \tag{$a2$}$$
$$(x \cdot y) \cdot z = x \cdot (y \cdot z). \tag{$a3$}$$

Note that the axiomatization of group theory has been modified in so that there exists a left identity and each element has a right inverse. Such an algebraic structure is called *(l, r) system* or *left group*. In the same manner, one can define a *(r, l) system* or *right group* by giving a right identity and a left inverse. We do not study (r, l) systems.

We list below the canonical term rewriting system for left group found after 20 seconds of computation. Sixteen new rules were derived, the method of Knuth and Bendix was used to give an orientation to the equations. Note the differences between this set and the canonical set for group theory listed in the previous section. In particular, 1 is no longer a right identity and x^{-1} is no longer a left inverse of x. However for all x, $x^{-1} \cdot (x \cdot y)$ is equal to y.

$$1 \cdot x \to x; \tag{r1}$$
$$x \cdot x^{-1} \to 1; \tag{r2}$$
$$(x \cdot y) \cdot z \to x \cdot (y \cdot z); \tag{r3}$$
$$1^{-1} \to 1; \tag{r4}$$
$$x \cdot (x^{-1} \cdot y) \to y; \tag{r5}$$
$$x \cdot 1 \to (x^{-1})^{-1}; \tag{r6}$$
$$(x^{-1})^{-1} \cdot y \to x \cdot y; \tag{r7}$$
$$x^{-1} \cdot (x \cdot y) \to y; \tag{r8}$$
$$((x^{-1})^{-1})^{-1} \to x^{-1}; \tag{r9}$$
$$(x \cdot y)^{-1} \to y^{-1} \cdot x^{-1}. \tag{r10}$$

Other presentations for (l, r) systems are studied by Knuth and Bendix. They also study the case of (r, l) systems.

6. Commutative Semigroups and Monoids.

We defined in example 3 the semigroup structure. We add now the commutativity axiom; that is, we consider the equational theory defined by the two following axioms:

$$(x \cdot y) \cdot z = x \cdot (y \cdot z); \tag{a1}$$
$$x \cdot y = y \cdot x. \tag{a2}$$

From now on at least one of the operators we shall consider will satisfy these two axioms. Thus we shall use the Peterson and Stickel's extension of Knuth and Bendix's algorithm. Forexample, if we declare $\cdot$ to be associative and commutative, the following set of one rule is a canonical term rewriting system for commutative monoid theory:

$$1 \cdot x \to x. \tag{r1}$$

Note that in this case, we do not have to consider the extended rule (the embedding of Peterson and Stickel):

$$1 \cdot x \cdot y \to x \cdot y,$$

for this rule is an instance of $(r1)$.
When the semigroup operation is commutative it is possible to embed the idempotency law in a canonical term rewriting system reduced to this single rule:

$$x \cdot x \to x. \tag{r1}$$

This time we have to consider also the extended rule:

$$x \cdot x \cdot y \rightarrow x \cdot y. \tag{er1}$$

One has to compare with example 3.

7. Abelian Groups.

We define abelian group theory by the following set of equations, where the additive notation is used:

$$x + y = y + x; \tag{a1}$$
$$(x + y) + z = x + (y + z); \tag{a2}$$
$$x + 0 = x; \tag{a3}$$
$$x + (-x) = 0. \tag{a4}$$

Equations ($a3$) and ($a4$) are proposed to the AC extension, where + is declared to be associative and commutative (equations ($a1$) and ($a2$)). We give below the complete development of this example, which can be compared with example 4. ($a3$) and ($a4$) are considered as left to right rewrite rules:

$$x + 0 \rightarrow x; \tag{r1}$$
$$x + (-x) \rightarrow 0. \tag{r2}$$

Note that we need to consider extended rules only for rule ($r2$):

$$x + (-x) + y \rightarrow y. \tag{er2}$$

The two rules are combined to derive:

$$-0 \rightarrow 0. \tag{r3}$$

Then two rules are deduced from the superposition of ($er2$) on itself:

$$-(-x) \rightarrow x; \tag{r4}$$
$$-(x + y) + y \rightarrow -x. \tag{r5}$$

And superposing ($er2$) on ($r5$):

$$-(x + y) \rightarrow (-x) + (-y). \tag{r6}$$

At this step rule (r5) is deleted and the remaining set of rewrite rules is shown to be canonical. We list below this set after renaming the rules:

$$x + 0 \to x; \tag{r1}$$

$$x + (-x) \to 0; \tag{r2}$$

$$-0 \to 0; \tag{r3}$$

$$-(-x) \to x; \tag{r4}$$

$$-(x + y) \to (-x) + (-y). \tag{r5}$$

This set was found in 18 seconds. Note that the orientation of the rules was given by the user. We now have to show the finite termination property. For this example, we do it by using the interpretation:

$$\begin{aligned}\chi(+) &= \lambda x_1 \ldots x_n . x_1 + \cdots + x_n,\\ \chi(-) &= \lambda x . x^2,\\ \chi(0) &= 2,\end{aligned}$$

over integers greater than 1.

Lankford has proposed to orient rule (r5) from right to left. In this case we obtain another canonical set of reductions for abelian groups. Rules (r1) to (r4) are the same, the others are:

$$(-x) + (-y) \to -(x + y); \tag{r5'}$$

$$-((-x) + y) \to x + (-y); \tag{r6'}$$

$$x + -(y + x) \to -y. \tag{r7'}$$

In this case we need to consider extended rules for rules (r5') and (r7'), that is:

$$(-x) + (-y) + z \to -(x + y) + z; \tag{er5'}$$

$$x + -(y + x) + z \to (-y) + z. \tag{er7'}$$

The finite termination property may be shown, using the interpretation:

$$\begin{aligned}\chi(+) &= \lambda x_1 \ldots x_n . x_1 + \cdots + x_n,\\ \chi(-) &= \lambda x . (x + 1),\\ \chi(0) &= 1,\end{aligned}$$

over integers greater than 1.

In all the following examples we shall use the first canonical term rewriting system of five rules as a precompiled set for abelian group theory.

8. Rings.

A *ring* is an abelian group written additively, with a binary operation noted $*$ such that:

$$x * (y + z) = (x * y) + (x * z); \tag{a5}$$

$$(x + y) * z = (x * z) + (y * z). \tag{a6}$$

These two axioms are the two distributivity laws. We propose these two equations to the program as left to right rewrite rules named ($r6$) and ($r7$). The first five rules are the canonical term rewriting system of the previous section. The program begins to superpose ($r1$) on ($r6$) leading to:

$$(x * 0) + (x * y) \rightarrow x * y. \tag{r8}$$

For this rule, we must consider also the extended rule:

$$(x * 0) + (x * y) + z \rightarrow (x * y) + z. \tag{er8}$$

Superposing ($er2$) on ($er8$) one can find immediately:

$$x * 0 = 0.$$

However, the program does not begin with this superposition and five rules are generated before:

$$(0 * x) + (y * x) \rightarrow y * x; \tag{r9}$$

$$(x * (-y)) + (x * y) \rightarrow x * 0; \tag{r10}$$

$$((-x) * y) + (x * y) \rightarrow 0 * y; \tag{r11}$$

$$-(x * y) + -(x * 0) \rightarrow -(x * y); \tag{r12}$$

$$-(x * 0) + (x * y) \rightarrow x * y. \tag{r13}$$

And finally the expected rule is found:

$$x * 0 \rightarrow 0. \tag{r14}$$

This rule is used to delete ($r8$), ($r12$) and ($r13$). Note that these last two rules have never participated in the derivation. ($r10$) is now replaced by:

$$(x * (-y)) + (x * y) \rightarrow 0. \tag{r15}$$

Two rules are now derived from $(er2)$ and $(er15)$:

$$-(x*(-y)) \to x*y; \qquad (r16)$$

$$x*(-y) \to -(x*y). \qquad (r17)$$

At this step rules $(r15)$ and $(r16)$ are deleted. After six new derivations where the rules symmetric to $(r14)$ and $(r17)$ are found, a canonical term rewriting system is obtained. We list it below after renaming. The first five rules are omitted.

$$x*(y+z) \to (x*y)+(x*z); \qquad (r6)$$

$$(x+y)*z \to (x*z)+(y*z); \qquad (r7)$$

$$x*0 \to 0; \qquad (r8)$$

$$x*(-y) \to -(x*y); \qquad (r9)$$

$$0*x \to 0; \qquad (r10)$$

$$(-x)*y \to -(x*y). \qquad (r11)$$

This computation took 60 seconds. The finite termination property may be shown using the interpretation:

$$\chi(+) = \lambda x_1 \ldots x_n . x_1 + \cdots + x_n + 5,$$
$$\chi(*) = \lambda x_1 \ldots x_n . x_1 \times \cdots \times x_n + 1,$$
$$\chi(-) = \lambda x . 2 \times (x+1),$$
$$\chi(0) = 2,$$

over integers greater than 2.

A canonical set for rings with unit (denoted by 1) is obtained by adding to the previous set the two new rewrite rules:

$$1*x \to x; \qquad (r12)$$

$$x*1 \to x. \qquad (r13)$$

A canonical set for associative ring with unit is obtained by adding the associativity law as a rewrite rule:

$$(x*y)*z \to x*(y*z). \qquad (r14)$$

If we declare that * is associative and commutative, some rewrite rules of the previous canonical set disappear and the new canonical term rewriting system for associative and commutative rings with unit is listed below after renaming. Rules $(r1)$ to $(r5)$ are omitted.

$$x*(y+z) \to (x*y)+(x*z); \qquad (r6)$$

$$x*0 \to 0; \qquad (r7)$$

$$x*(-y) \to -(x*y); \qquad (r8)$$

$$x*1 \to x. \qquad (r9)$$

Let us now study some particular rings.

9. Anticommutative Rings and Lie Rings.

In this section we present an example that is out of the scope of the present study since an equation we cannot handle is derived.

Definition. We call *Lie ring* a ring whose multiplicative law, denoted by [,], verifies:

$$\forall x \quad [x, x] = 0,$$

and Jacobi's identity:

$$\forall x, y, z \quad [[x, y], z] + [[y, z], x] + [[z, x], y] = 0.$$

We would like to find a canonical term rewriting system for Lie rings, but the equation $[x, x] = 0$ does not allow us to do it. Indeed, assume we give this equation to the program together with the precompiled set of 11 rewrite rules for rings found in the previous section (where the multiplication law is renamed). We list below the first rewrite rules derived:

$$[x, x] \to 0; \qquad (r12)$$

$$[x, y] + [y, x] \to 0; \qquad (r13)$$

$$-[x, y] \to [y, x]; \qquad (r14)$$

$$[[x, y], [y, x]] \to 0; \qquad (r15)$$

$$[[x, y], z] + [[y, x], z] \to 0; \qquad (r16)$$

$$[x, (-y)] \to [y, x]; \qquad (r17)$$

$$[(-x), y] \to [y, x]. \qquad (r18)$$

Then we stop the computation when the following equation is proposed:

$$[x, [y, z]] = [[z, y], x],$$

since it is impossible to give an orientation without losing the finite termination property. For instance, if we choose left to right orientation, that is:

$$[x, [y, z]] \to [[z, y], x], \qquad (r19)$$

and if we apply this rule to the term:

$$[[x, y], [y, x]],$$

we obtain term:

$$[[x, y], [x, y]],$$

to which we apply rule (r19) three times more:

$$[[y,x],[x,y]]$$

$$[[y,x],[y,x]]$$

$$[[x,y],[y,x]]$$

that is:

$$[[x,y],[y,x]] \rightarrow^{+} [[x,y],[y,x]]$$

where $\rightarrow^{+}$ is the transitive closure of $\rightarrow$. The other orientation leads us to an analogous result. If someone could handle the following property of Lie brackets:

$$[x,[y,x]] = [[x,y],x],$$

as we do with commutativity and associativity, this trouble would be eliminated.

10. A-modules.

Since we are not able to deal with conditional rewrite rules, we cannot study algebraic structures like fields or vector spaces on a field. However, we can study stuctures like A-modules, that is vector spaces on an associative ring with unit A.
Definition. Let A be an associative ring with unit:

$$A = (+, 0, *, 1).$$

We call *A-module* and we write M, the algebraic structure defined by:

- an internal law, mapping from $M \times M$ to M, denoted by $\oplus$:

$$(x,y) \mapsto x \oplus y,$$

- an external law, mapping from $A \times M$ to M, denoted by $\cdot$:

$$(\alpha, x) \mapsto \alpha \cdot x,$$

with the following properties:

1. M associated with $\oplus$ is an abelian group. The identity element is denoted by Ω, the inverse of x is denoted by $I(x)$.
2. $\forall(\alpha,\beta) \in A^2$, $\forall x \in M$:

$$\alpha \cdot (\beta \cdot x) = (\alpha * \beta) \cdot x;$$

$$1 \cdot x = x.$$

3. $\forall(\alpha,\beta) \in A^2$, $\forall(x,y) \in M^2$:

$$(\alpha + \beta) \cdot x = (\alpha \cdot x) \oplus (\beta \cdot x);$$

$$\alpha \cdot (x \oplus y) = (\alpha \cdot x) \oplus (\alpha \cdot y).$$

For instance, every commutative ring with unit A is an A-module on itself. As another example, consider the $(n \times n)$ matrices with elements in a commutative ring with unit A. We call this set $M_n(A)$; it is easy to show it is an A-module.

In order to reduce the number of rewrite rules used in the following derivation, we will assume now that the ring A is commutative. However, in each case where a canonical term rewriting system will be found another could be found, without the commutativity property.

Following the definition, we can now give an axiomatisation for free A-modules. First we take the axiomatization for free commutative rings given in section 8; we introduce the equations of section 7 defining the abelian group structure of M and the four equations:

$$\alpha \cdot (\beta \cdot x) = (\alpha * \beta) \cdot x; \qquad (a1)$$

$$1 \cdot x = x; \qquad (a2)$$

$$(\alpha \cdot x) \oplus (\beta \cdot x) = (\alpha + \beta) \cdot x; \qquad (a3)$$

$$\alpha \cdot (x \oplus y) = (\alpha \cdot x) \oplus (\alpha \cdot y). \qquad (a4)$$

We try now to build a canonical term rewriting system. First, it is easy to show that the following set is canonical (since it is composed of two canonical sets, and obviously no superposition is possible between the rewrite rules of these two sets):

$$\alpha + 0 \rightarrow \alpha; \qquad (r1)$$

$$\alpha + (-\alpha) \rightarrow 0; \qquad (r2)$$

$$-0 \rightarrow 0; \qquad (r3)$$

$$-(-\alpha) \rightarrow \alpha; \qquad (r4)$$

$$-(\alpha + \beta) \rightarrow (-\alpha) + (-\beta); \qquad (r5)$$

$$1 * \alpha \rightarrow \alpha; \qquad (r6)$$

$$\alpha * (\beta + \gamma) \rightarrow (\alpha * \beta) + (\alpha * \gamma); \qquad (r7)$$

$$\alpha * 0 \rightarrow 0; \qquad (r8)$$

$$\alpha * (-\beta) \rightarrow -(\alpha * \beta); \qquad (r9)$$

$$x \oplus \Omega \rightarrow x; \qquad (r10)$$

$$x \oplus I(x) \rightarrow \Omega; \qquad (r11)$$

$$I(\Omega) \rightarrow \Omega; \qquad (r12)$$

$$I(I(x)) \rightarrow x; \qquad (r13)$$

$$I(x \oplus y) \rightarrow I(x) \oplus I(y). \qquad (r14)$$

$+$, $*$ and $\oplus$ are declared associative and commutative. From now on we do not explicit the extended rules.

Remark To improve readability, we denote elements in A by greek letters and elements in M by roman letters. However, *our language of terms is not typed*, although we would get exactly the same superpositions in a typed language.

In a second step, we introduce this canonical set together with equations (a1) to (a4). We orient these four equations from left to right: this corresponds basically to the intuition we have of a normal form in such a theory, that is:

$$\sum_{i \in I} a_i \cdot x_i \quad where \quad \forall i,j \quad i \neq j \quad x_i \neq x_j.$$

The four corresponding rules are called respectively (r15), (r16), (r17) and (r18):

$$\alpha \cdot (\beta \cdot x) \rightarrow (\alpha * \beta) \cdot x; \qquad (r15)$$

$$1 \cdot x \rightarrow x; \qquad (r16)$$

$$(\alpha \cdot x) \oplus (\beta \cdot x) \rightarrow (\alpha + \beta) \cdot x; \qquad (r17)$$

$$\alpha \cdot (x \oplus y) \rightarrow (\alpha \cdot x) \oplus (\alpha \cdot y). \qquad (r18)$$

The first equations derived are quite interesting, and we follow the same intuition to orient them:

$$x \oplus (\alpha \cdot x) \rightarrow (1 + \alpha) \cdot x; \qquad (r19)$$

$$x \oplus x \rightarrow (1 + 1) \cdot x; \qquad (r20)$$

$$\alpha \cdot \Omega \rightarrow \Omega; \qquad (r21)$$

$$0 \cdot x \rightarrow \Omega. \qquad (r22)$$

After these four rules the system derives equations like:

$$((1+1) \cdot x) \oplus ((1+1) \cdot I((1+1) \cdot x)) = I((1+1) \cdot x).$$

These rules become bigger and bigger and we can no more believe in the convergence of such a system. The problem would be solved if the program could find equation:

$$I(x) = (-1) \cdot x.$$

However, we do not want to give it the solution and we try another way: we restart the program with equations (a1) to (a4), but this time we orient equation (a3) from right to left, that is (r17) becomes (r17'):

$$(\alpha + \beta) \cdot x \rightarrow (\alpha \cdot x) \oplus (\beta \cdot x). \qquad (r17')$$

In this case, we obtain a canonical set consisting of rules ($r1$) to ($r16$) and rules ($r17'$), ($r18$), together with the following new rules:

$$0 \cdot x \rightarrow \Omega; \qquad (r19')$$

$$\alpha \cdot \Omega \rightarrow \Omega; \qquad (r20')$$

$$(-\alpha) \cdot x \rightarrow \alpha \cdot I(x); \qquad (r21')$$

$$(\alpha \cdot x) \oplus (\alpha \cdot I(x)) \rightarrow \Omega; \qquad (r22')$$

$$I(\alpha \cdot x) \rightarrow \alpha \cdot I(x). \qquad (r23')$$

We shall use this canonical term rewriting system for A-modules in examples 12, 13, 14. However, the normal form given by this canonical set is not the one we were looking for. But now we know that $I(x) = (-1) \cdot x$ is an equation in our theory (put $\alpha = 1$ in ($r21'$)). Thus, we can restart our program with rules ($r1$) to ($r18$) and the new rule:

$$I(x) \rightarrow (-1) \cdot x. \qquad (r19)$$

Then, we obtain a new canonical term rewriting system for free A-modules listed below after renaming:

$$\alpha + 0 \rightarrow 0; \qquad (r1)$$

$$\alpha + (-\alpha) \rightarrow 0; \qquad (r2)$$

$$-0 \rightarrow 0; \qquad (r3)$$

$$-(-\alpha) \rightarrow \alpha; \qquad (r4)$$

$$-(\alpha + \beta) \rightarrow (-\alpha) + (-\beta); \qquad (r5)$$

$$1 * \alpha \rightarrow \alpha; \qquad (r6)$$

$$\alpha * (\beta + \gamma) \rightarrow (\alpha * \beta) + (\alpha * \gamma); \qquad (r7)$$

$$\alpha * 0 \rightarrow 0; \qquad (r8)$$

$$\alpha * (-\beta) \rightarrow -(\alpha * \beta); \qquad (r9)$$

$$x \oplus \Omega \rightarrow x; \qquad (r10)$$

$$\alpha \cdot (\beta \cdot x) \rightarrow (\alpha * \beta) \cdot x; \qquad (r11)$$

$$1 \cdot x \rightarrow x; \qquad (r12)$$

$$(\alpha \cdot x) \oplus (\beta \cdot x) \rightarrow (\alpha + \beta) \cdot x; \qquad (r13)$$

$$\alpha \cdot (x \oplus y) \rightarrow (\alpha \cdot x) \oplus (\alpha \cdot y); \qquad (r14)$$

$$x \oplus (\alpha \cdot x) \rightarrow (1 + \alpha) \cdot x; \qquad (r15)$$

$$x \oplus x \rightarrow (1 + 1) \cdot x; \qquad (r16)$$

$$\alpha \cdot \Omega \rightarrow \Omega; \qquad (r17)$$

$$0 \cdot x \rightarrow \Omega; \qquad (r18)$$

$$I(x) \rightarrow (-1) \cdot x. \qquad (r19)$$

This canonical set gives us the expected normal form. Note that we will use both sets in the following examples.

11. A-bimodules.

The A-modules studied in example 10 are sometimes called left A-modules. Indeed, one can symmetrically study the structure called *right A-modules* defined in an analogous way. This time the external law is a mapping from $M \times A$ to M, denoted by $\circ$ with the following properties:

2′. $\forall(\alpha,\beta) \in A^2, \forall x \in M$:

$$(x \circ \alpha) \circ \beta = x \circ (\alpha * \beta);$$
$$x \circ 1 = x.$$

3′. $\forall(\alpha,\beta) \in A^2, \forall(x,y) \in M^2$:

$$x \circ (\alpha + \beta) = (x \circ \alpha) \oplus (x \circ \beta);$$
$$(x \oplus y) \circ \alpha = (x \circ \alpha) \oplus (y \circ \alpha).$$

We are interested in this section in the case where M is a left and right A-module such that the following identity holds:

$$(\alpha \cdot x) \circ \beta = \alpha \cdot (x \circ \beta).$$

Such a structure is called *two-sided A-module* or *A-bimodule*. One can easily see that a normal form of the kind we were studying in section 10 (second canonical term rewriting system) cannot exist in a A-bimodule. So we will use the first canonical term rewriting system found for left A-modules. The fourteen first rules of this set are listed at the beginning of section 10. We list below the remaining rules and give them names ($r15$) to ($r23$):

$$\alpha \cdot (\beta \cdot x) \to (\alpha * \beta) \cdot x; \qquad (r15)$$
$$1 \cdot x \to x; \qquad (r16)$$
$$(\alpha + \beta) \cdot x \to (\alpha \cdot x) \oplus (\beta \cdot x); \qquad (r17)$$
$$\alpha \cdot (x \oplus y) \to (\alpha \cdot x) \oplus (\alpha \cdot y); \qquad (r18)$$
$$0 \cdot x \to \Omega; \qquad (r19)$$
$$\alpha \cdot \Omega \to \Omega; \qquad (r20)$$
$$(-\alpha) \cdot x \to \alpha \cdot I(x); \qquad (r21)$$
$$(\alpha \cdot x) \oplus (\alpha \cdot I(x)) \to \Omega; \qquad (r22)$$
$$I(\alpha \cdot x) \to \alpha \cdot I(x). \qquad (r23)$$

Moreover we have to give the rules corresponding to the right A-module structure of M:

$$(x \circ \alpha) \circ \beta \to x \circ (\alpha * \beta); \tag{r24}$$

$$x \circ 1 \to x; \tag{r25}$$

$$x \circ (\alpha + \beta) \to (x \circ \alpha) \oplus (x \circ \beta); \tag{r26}$$

$$(x \oplus y) \circ \alpha \to (x \circ \alpha) \oplus (y \circ \alpha); \tag{r27}$$

$$x \circ 0 \to \Omega; \tag{r28}$$

$$\Omega \circ \alpha \to \Omega; \tag{r29}$$

$$x \circ (-\alpha) \to I(x) \circ \alpha; \tag{r30}$$

$$(x \circ \alpha) \oplus (I(x) \circ \alpha) \to \Omega; \tag{r31}$$

$$I(x \circ \alpha) \to I(x) \circ \alpha. \tag{r32}$$

And finally we introduce the previous equation as a left to right rewrite rule:

$$(\alpha \cdot x) \circ \beta \to \alpha \cdot (x \circ \beta). \tag{r33}$$

A new rule is then derived from rules (r31) and (r23):

$$\alpha \cdot (I(x) \circ \beta) \oplus \alpha \cdot (x \circ \beta) \to \Omega. \tag{r34}$$

This set of 34 rules is a canonical term rewriting system for free A-bimodules.

12. A-rings.

Definition. Let A be an associative ring with unit, then a ring $M = (\oplus, \Omega, \otimes)$ is called a *A-ring* if it is a A-bimodule such that $\forall x, y \in M$, $\forall \alpha \in A$:

$$(\alpha \cdot x) \otimes y = \alpha \cdot (x \otimes y);$$
$$(x \circ \alpha) \otimes y = x \otimes (\alpha \cdot y);$$
$$x \otimes (y \circ \alpha) = (x \otimes y) \circ \alpha.$$

Note that A-rings are rings with a ring as coefficient domain. A-algebras are a particular case of A-rings, they will be studied in the next section. If M is an associative ring with unit and A is any subring of M, then M is an A-ring.

For this example we list only the canonical set obtained. Note that the computation looks like the simpler one we will give in the next section for A-algebras. The first 34 rules are the rules leading to a canonical set for A-bimodules and are omitted. Then we list the rules corresponding to the ring structure of M:

$$x \otimes (y \oplus z) \to (x \otimes y) \oplus (x \otimes z); \tag{r35}$$

$$(x \oplus y) \otimes z \to (x \otimes z) \oplus (y \otimes z); \tag{r36}$$

$$x \otimes \Omega \to \Omega; \tag{r37}$$

$$x \otimes I(y) \to I(x \otimes y); \tag{r38}$$

$$\Omega \otimes x \to \Omega; \tag{r39}$$

$$I(x) \otimes y \to I(x \otimes y). \tag{r40}$$

And finally the rules typical of the A-ring structure:

$$(a \cdot x) \otimes y \to a \cdot (x \otimes y); \tag{r41}$$

$$(x \circ a) \otimes y \to x \otimes (a \cdot y); \tag{r42}$$

$$x \otimes (y \circ a) \to (x \otimes y) \circ a; \tag{r43}$$

$$x \otimes (a \cdot I(y)) \to I(x \otimes (a \cdot y)); \tag{r44}$$

$$x \otimes (a \cdot (y \circ \beta)) \to (x \otimes (a \cdot y)) \circ \beta. \tag{r45}$$

In the next section we study a particular case of A-rings.

13. A-algebras.

Definition. Let A be a commutative ring with unit. We call *A-algebra* a pair $\langle M, \otimes \rangle$ where M is an A-module and $\otimes$ is a bilinear mapping from $M \times M$ to M. An A-algebra will be said to be associative (resp. commutative) iff $\otimes$ has this property.

Another way to define a A-algebra is to say that M is a A-ring such that A is commutative and:

$$\forall x \in M, \quad \forall a \in A \qquad a \cdot x = x \circ a.$$

For instance, consider the A-module of $(n \times n)$ matrices $M_n(A)$ and the operation $\otimes$ of matrix product. It is easy to show that this operation is bilinear and associative (but not commutative). Thus $\langle M_n(A), \otimes \rangle$ is an associative A-algebra.

If we add the following equations to the one defining an A-module, we obtain an axiomatization for free A-algebras:

$$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z); \tag{a1}$$

$$(x \oplus y) \otimes z = (x \otimes z) \oplus (y \otimes z); \tag{a2}$$

$$(a \cdot x) \otimes (\beta \cdot y) = (a * \beta) \cdot (x \otimes y); \tag{a3}$$

$$(x \otimes y) \otimes z = x \otimes (y \otimes z). \tag{a4}$$

We give these four rules to the program together with the canonical set for A-modules discovered in section 10 (we begin with the second one). These four rules are oriented from left to right; we name them ($r20$) to ($r23$). Using ($r22$) and ($r12$), the following equation is found:

$$x\otimes(\alpha\cdot y)=\alpha\cdot(x\otimes y),$$

which is a way of expressing linearity of $\otimes$ in its second argument. This equation is oriented from left to right leading to rewrite rule ($r24$). Now rule ($r22$) needs to be rewritten, thus is replaced by:

$$\alpha\cdot((\beta\cdot x)\otimes y)\rightarrow(\alpha*\beta)\cdot(x\otimes y). \tag{$r25$}$$

Linearity of $\otimes$ in its first argument is now found by superposing rules ($r25$) and ($r12$), leading to rewrite rule ($r26$):

$$(\alpha\cdot x)\otimes y\rightarrow\alpha\cdot(x\otimes y), \tag{$r26$}$$

and ($r25$) is deleted. Note that this is equivalent to specify bilinearity of $\otimes$ with ($r22$) or with the two equations ($r24$) and ($r26$).

A new equation is then derived using rules ($r17$) and ($r24$); with a convenient orientation:

$$\alpha\cdot(x\otimes\Omega)\rightarrow x\otimes\Omega, \tag{$r27$}$$

and superposing this rule with ($r18$):

$$x\otimes\Omega\rightarrow\Omega. \tag{$r28$}$$

At this step, rule ($r27$) is deleted and the program derives in the same manner:

$$\Omega\otimes x\rightarrow\Omega. \tag{$r30$}$$

This set of 26 rules is a canonical set of reductions. We list below the corresponding rules after renaming (rules ($r1$) to ($r19$) are omitted):

$$x\otimes(y\oplus z)\rightarrow(x\otimes y)\oplus(x\otimes z); \tag{$r20$}$$

$$(x\oplus y)\otimes z\rightarrow(x\otimes z)\oplus(y\otimes z); \tag{$r21$}$$

$$(x\otimes y)\otimes z\rightarrow x\otimes(y\otimes z); \tag{$r22$}$$

$$x\otimes(\alpha\cdot y)\rightarrow\alpha\cdot(x\otimes y); \tag{$r23$}$$

$$(\alpha\cdot x)\otimes y\rightarrow\alpha\cdot(x\otimes y); \tag{$r24$}$$

$$x\otimes\Omega\rightarrow\Omega; \tag{$r25$}$$

$$\Omega\otimes x\rightarrow\Omega. \tag{$r26$}$$

We then add the commutativity axiom for $\otimes$ to the axiomatization given above, that is $\otimes$ is declared to be associative and commutative. The development of the program is quite the same. We give the canonical set obtained (rules ($r1$) to ($r19$) are omitted):

$$x \otimes (y \oplus z) \to (x \otimes y) \oplus (x \otimes z); \qquad (r20)$$

$$x \otimes (a \cdot y) \to a \cdot (x \otimes y); \qquad (r21)$$

$$x \otimes \Omega \to \Omega. \qquad (r22)$$

If we use now the second definition of an A-algebra as a particular case of A-ring, we can deduce from the canonical set of the previous section a canonical set for A-algebras. Rules ($r1$) to ($r23$) are the same and are omitted; we list below the other new rules:

$$x \otimes (y \oplus z) \to (x \otimes y) \oplus (x \otimes z); \qquad (r24)$$

$$(x \oplus y) \otimes z \to (x \otimes z) \oplus (y \otimes z); \qquad (r25)$$

$$(x \otimes y) \otimes z \to x \otimes (y \otimes z); \qquad (r26)$$

$$x \otimes (a \cdot y) \to a \cdot (x \otimes y); \qquad (r27)$$

$$x \otimes \Omega \to \Omega; \qquad (r28)$$

$$x \otimes I(y) \to I(x \otimes y); \qquad (r29)$$

$$(a \cdot x) \otimes y \to a \cdot (x \otimes y); \qquad (r30)$$

$$\Omega \otimes x \to \Omega; \qquad (r31)$$

$$I(x) \otimes y \to I(x \otimes y). \qquad (r32)$$

One can now deduce without difficulty a canonical set in the case where $\otimes$ is commutative:

$$x \otimes (y \oplus z) \to (x \otimes y) \oplus (x \otimes z); \qquad (r24)$$

$$x \otimes (a \cdot y) \to a \cdot (x \otimes y); \qquad (r25)$$

$$x \otimes \Omega \to \Omega; \qquad (r26)$$

$$x \otimes I(y) \to I(x \otimes y). \qquad (r27)$$

14. Lattices.

Besides associativity and commutativity of $\cup$ and $\cap$, we give the two absorption laws. That is, after a convenient orientation:

$$x \cap (x \cup y) \to x; \qquad (r1)$$

$$x \cup (x \cap y) \to x. \qquad (r2)$$

The idempotency laws for $\cup$ and $\cap$ follow by superposition of these two rules:

$$x \cup x \to x; \qquad (r3)$$

$$x \cap x \to x. \qquad (r4)$$

Then many consequences of rules ($r1$) and ($r2$) are found. We list some of them below:

$$(((x \cap y) \cup z) \cap x) \cup (x \cap y) \rightarrow ((x \cap y) \cup z) \cap x; \qquad (r5)$$

$$(((x \cup y) \cap z) \cup x) \cap (x \cup y) \rightarrow ((x \cup y) \cap z) \cup x; \qquad (r6)$$

$$((x \cup y) \cap z) \cup (z \cap x) \rightarrow (x \cup y) \cap z; \qquad (r7)$$

$$((x \cup y) \cap z) \cup (z \cap u \cap x) \rightarrow (x \cup y) \cap z. \qquad (r8)$$

Then some consequences of rule ($r3$) alone:

$$(((x \cup y) \cap z) \cup u) \cap z \cap x \rightarrow z \cap x; \qquad (r17)$$

$$((((x \cap y) \cup z) \cap y) \cup u) \cap x \cap y \rightarrow x \cap y. \qquad (r18)$$

We stop the computation at this point since no more hope is allowed.

Two identities are particularly important when dealing with lattices: the modular identity and the distributive identity. We shall see in the next section how the program finds a canonical form for distributive lattices. With the modular identity:

$$(x \cap y) \cup (x \cap z) = x \cap (y \cup (x \cap z)),$$

many rules are derived as for lattices, but no canonical term rewriting system.

15. Distributive Lattices.

This time we give associativity and commutativity of ∪ and ∩, the two absorption laws and one of the two distributivity laws, that is:

$$x \cap (x \cup y) \rightarrow x; \qquad (r1)$$

$$x \cup (x \cap y) \rightarrow x; \qquad (r2)$$

$$x \cup (y \cap z) \rightarrow (x \cup y) \cap (x \cup z). \qquad (r3)$$

Rule ($r2$) is immediately rewritten using the distributivity law, and is replaced by:

$$(x \cup y) \cap (x \cup x) \rightarrow x. \qquad (r4)$$

The idempotency law for ∪ is derived using rules ($r4$) and ($r1$):

$$x \cup x \rightarrow x, \qquad (r5)$$

then ($r4$) is deleted and the idempotency law for ∩ is added from ($r5$) and ($r1$). After five minutes of computation, without generating any rule, this set of rules is declared to be a canonical term rewriting system:

$$x \cap (x \cup y) \rightarrow x; \qquad (r1)$$

$$x \cup (y \cap z) \rightarrow (x \cup y) \cap (x \cup z); \qquad (r3)$$

$$x \cup x \rightarrow x; \qquad (r5)$$

$$x \cap x \rightarrow x. \qquad (r6)$$

It is easy to show the finite termination property, using the interpretation:

$$\chi(\cap) = \lambda x_1 \ldots x_n . x_1 + \cdots + x_n + 1,$$

$$\chi(\cup) = \lambda x_1 \ldots x_n . x_1 \times \cdots \times x_n,$$

over integers greater than 1.

Using this canonical set, we are now able to show the second distributivity law. We have the following sequence of reductions. First using ($r3$):

$$(x \cap y) \cup (x \cap z) \rightarrow ((x \cap y) \cup x) \cap ((x \cap y) \cup z),$$

then using again rule ($r3$) twice:

$$(x \cup x) \cap (y \cup x) \cap (x \cup z) \cap (y \cup z),$$

then, by rule ($r5$):

$$x \cap (y \cup x) \cap (x \cup z) \cap (y \cup z),$$

finally, using twice rule ($r1$):

$$x \cap (y \cup z),$$

proving the second distributivity law. This example is from Peterson and Stickel (77).

16. Boolean Algebras.

Using the canonical term rewriting system for free distributive lattices we would like now to find a normal form for Boolean algebras. Unfortunately we will not achieve this aim. We give below some of our experiments with Boolean algebras. Rules ($r1$) to ($r4$) are the rules for distributive lattices. An axiomatization for Boolean algebras is obtained by adding the following equations as left to right rewrite rules:

$$x \cup 0 \rightarrow x; \tag{r5}$$

$$x \cup \overline{x} \rightarrow 1; \tag{r6}$$

$$x \cap 1 \rightarrow x; \tag{r7}$$

$$x \cap \overline{x} \rightarrow 0. \tag{r8}$$

Four simple rules are then derived:

$$\overline{0} \rightarrow 1; \tag{r9}$$

$$\overline{1} \rightarrow 0; \tag{r10}$$

$$0 \cap x \rightarrow 0; \tag{r11}$$

$$1 \cup x \rightarrow 1. \tag{r12}$$

Then begins a quite lengthy computation, leading to rewrite rules:

$$(\overline{(x \cap y)} \cup y) \cap (\overline{(x \cap y)} \cup y) \rightarrow 1; \tag{r13}$$

$$(x \cup \overline{y}) \cap (x \cup y) \rightarrow x; \tag{r14}$$

$$(x \cup \overline{(y \cup z)}) \cap (x \cup y) \cap (x \cup z) \rightarrow x; \tag{r15}$$

$$(x \cup \overline{(y \cup x)}) \cap (x \cup y) \rightarrow x; \tag{r16}$$

$$(x \cup y \cup \overline{(y \cup z)}) \cap (x \cup y \cup z) \rightarrow x \cup y. \tag{r17}$$

Most of these rules would be reduced if the program could find de Morgan's laws. Two new rules are then derived from ($r14$) and ($r6$):

$$x \cup \overline{\overline{x}} \to x; \qquad (r18)$$

$$\overline{\overline{x}} \to x. \qquad (r19)$$

At this step, rule ($r18$) is deleted. Now a new lengthy computation begins. We list the first rules derived:

$$x \cup \overline{(\overline{(x \cup y)} \cup y)} \to x; \qquad (r20)$$

$$\overline{(x \cup y)} \cup x \cup \overline{y} \to \overline{(x \cup y)} \cup x; \qquad (r21)$$

$$\overline{(\overline{x} \cup y)} \cup x \cup y \to \overline{(\overline{x} \cup y)} \cup y. \qquad (r22)$$

The computation continues and we stop the program after rule ($r30$) because no more hope is allowed the program find de Morgan's laws soon.

We then enter de Morgan's law by hand that is, we begin a new computation in giving rules ($r1$) to ($r12$) and rule ($r19$) to which we add the two following:

$$\overline{x \cup y} \to \overline{x} \cap \overline{y}; \qquad (r14')$$

$$\overline{x \cap y} \to \overline{x} \cup \overline{y}. \qquad (r15')$$

The first rules derived are:

$$x \cap y \cap (\overline{x} \cup \overline{y}) \to 0; \qquad (r16')$$

$$(x \cup y) \cap \overline{x} \cap \overline{y} \to 0. \qquad (r17')$$

And the following shows the process would never terminate:

$$x \cap y \cap z \cap (\overline{x} \cup \overline{y} \cup \overline{z}) \to 0; \qquad (r18')$$

$$(x \cup y \cup z) \cap \overline{x} \cap \overline{y} \cap \overline{z} \to 0. \qquad (r19')$$

Note that other orientations one can give to de Morgan's laws lead to the same result. In order to avoid this generation of an infinite set of rules, we try a new run by adding the two following rules which reduce rules ($r16'$) to ($r19'$):

$$(x \cup y) \cap \overline{x} \to y \cap \overline{x};$$

$$(\overline{x} \cup y) \cap x \to x \cap y.$$

But again the program begins to enumerate an infinite set of rules:

$$(x \cup \overline{y}) \cap (x \cup y) \to x;$$

$$(x \cup \overline{y}) \cap (x \cup \overline{z}) \cap (x \cup y \cup z) \to x.$$

Note that the main difficulty encountered by the program is when dealing with particular instances of the second distributivity law (the one we do not give to the program as a rewrite rule).

As noted by Peterson and Stickel, the trouble here comes from the fact that the normal form found by the program is the form of conjunctions of prime implicants. As is well known, this normal form is not unique.

17. R. Milner's Theory of Nondeterministic Machines.

We consider the following equational theory:

$$x + x = x; \tag{a1}$$

$$x + 0 = x; \tag{a2}$$

$$T(x) + x = T(x); \tag{a3}$$

$$T(x + T(y)) = T(x + y) + T(y); \tag{a4}$$

$$L(x + T(y)) = L(x + y) + L(y); \tag{a5}$$

$$(x + y) + z = x + (y + z); \tag{a6}$$

$$x + y = y + x. \tag{a7}$$

This is the theory considered by R. Milner for axiomatizing the behavior of finite non-deterministic machines. (Nil is replaced by 0, and we assume $\Lambda = \{L\}$).

We process this theory with $+$ declared associative and commutative; equations $(a1)$ to $(a5)$ are input as left to right rewrite rules called $(r1)$ to $(r5)$. The system generates the further rewrite rules:

$$T(x + y) + x \rightarrow T(x + y); \tag{r6}$$

$$T(T(x)) \rightarrow T(x); \tag{r7}$$

$$L(T(x)) \rightarrow L(x). \tag{r8}$$

Note that rule $(r6)$ is derived from $(r2)$ and $(r3)$. The program then stops, stating that rules $(r1)$ to $(r8)$ form a canonical set. The finite termination property is shown using the interpretation:

$$\chi(+) = \lambda x_1 \ldots x_n . x_1 + \cdots + x_n,$$
$$\chi(0) = 2,$$
$$\chi(T) = \lambda x . x^2,$$
$$\chi(L) = \lambda x . x^2,$$

over integers greater than 1. The argument can easily be extended to:

$$\Lambda = \{L_1, \ldots, L_n\},$$

by adding appropriate instances of $(r5)$ and $(r8)$.

R. Milner has proposed to replace rules (r4) and (r5) by the following:

$$T(x + T(y)) + T(y) \rightarrow T(x + T(y)); \qquad (r4')$$

$$L(x + T(y)) + L(y) \rightarrow L(x + T(y)). \qquad (r5')$$

All others rules of the previous canonical set being preserved. Rule (r4′) is first deleted since it is rewritten by rule (r6). We give below the first three rules generated:

$$x + T(T(x) + y) \rightarrow T(y + T(x)); \qquad (r9)$$

$$x + T(y + T(z + x)) \rightarrow T(y + T(x + z)); \qquad (r10)$$

$$x + T(y + T(T(x) + z)) \rightarrow T(y + T(T(x) + z)). \qquad (r11)$$

It is useless to go further since the process would never terminate; let us write rule (r9) as:

$$x_0 + T(x_1 + T(x_0)) \rightarrow T(x_1 + T(x_0)),$$

and rule (r11) (which is derived from (r9) and (r5)) as:

$$x_0 + T(x_1 + T(x_2 + T(x_0))) \rightarrow T(x_1 + T(x_2 + T(x_0))).$$

It is easy to see from the construction of these two rules that an infinite sequence of rewrite rules will be derived:

$$x_0 + T(x_1 + T(x_2 + \cdots + T(x_n + T(x_0))\cdots)) \rightarrow T(x_1 + T(x_2 + \cdots + T(x_n + T(x_0))\cdots)).$$

18. Arithmetic Theories.

P. Degano and F. Sirovich (79) have solved the equivalence problem for a subclass of primitive recursive functions by using the following term rewriting system:

$$0 + x \rightarrow x; \qquad (r1)$$

$$0 \times x \rightarrow 0; \qquad (r2)$$

$$S(x) + y \rightarrow S(x + y); \qquad (r3)$$

$$S(x) \times y \rightarrow (x \times y) + y; \qquad (r4)$$

$$x \times (y + z) \rightarrow (x \times y) + (x \times z). \qquad (r5)$$

where + and × are assumed to be associative and commutative. They have shown by hand that this set is canonical and it has been confirmed by our program and Stickel's. Furthermore, Stickel has shown how to extend this set by adding exponentiation. The new canonical set is listed below (rules $(r1)$ to $(r5)$ are omitted):

$$x^0 \rightarrow S(0); \tag{r6}$$

$$S(0)^x \rightarrow S(0); \tag{r7}$$

$$x^{S(v)} \rightarrow x \times x^v; \tag{r8}$$

$$x^v \times z^v \rightarrow (x \times z)^v; \tag{r9}$$

$$x^{(v+s)} \rightarrow x^v \times x^s. \tag{r10}$$

Another interesting use of these canonical sets for arithmetic theories may be found in Huet and Hullot (80) in the context of proofs by induction.

19. R.J. Popplestone's Theory of Robot's Movements.

This example has been proposed by R.J. Popplestone. We consider three kinds of transformations or "movements"that can affect a solid. Essentially, we consider the movements of a cylinder on an horizontal axis, which is its principal axis; the cylinder can turn around this axis say $x'x$, the sign of the rotation is determined by the reference's axes $(y'y, z'z, x'x)$. One can change the sign of the rotation in considering $(yy', z'z, xx')$ instead. Moreover, the cylinder can move along $x'x$ and $x'x$ itself can move in the three-dimensional space but it must remain horizontal. These transformations are represented well by matrices (4×4) in homogeneous coordinates. To turn of θ around $x'x$ we have to apply operator $T(\theta)$:

$$T(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

to move along one of the axes we have to apply the translation operator:

$$Tr(x, y, z) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ x & y & z & 1 \end{pmatrix},$$

and to switch from $(y'y, z'z, x'x)$ to $(yy', z'z, xx')$ we have to apply the symmetry $\mathcal{M}$:

$$\mathcal{M} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Between these transformations, we have the following relations, which are easy to check where $*$ is the product of matrices and $\mathcal{I}$ is the identity matrix:

$$Tr(x,y,z) * Tr(u,v,w) = Tr(x+u, y+v, z+w); \tag{a1}$$

$$T(\theta) * T(\theta') = T(\theta + \theta'); \tag{a2}$$

$$\mathcal{M} * \mathcal{M} = \mathcal{I}; \tag{a3}$$

$$T(\theta) * Tr(x,0,0) = Tr(x,0,0) * T(\theta); \tag{a4}$$

$$T(\theta) * \mathcal{M} = \mathcal{M} * T((-\theta)); \tag{a5}$$

$$Tr(x,y,z) * \mathcal{M} = \mathcal{M} * Tr((-x),(-y),z). \tag{a6}$$

Matrices will be denoted by script letters, that is $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$, the inverse of matrix $\mathcal{X}$ will be denoted by $\mathcal{X}^{-1}$. The space of (4×4) matrices will be considered as a nonabelian group, and the space of scalars as an abelian group. We are interested in finding a canonical form for the corresponding equational theory.

First we introduce the canonical set for nonabelian groups in order to deal with general operations on matrices:

$$\mathcal{I} * \mathcal{X} \rightarrow \mathcal{X}; \tag{r1}$$

$$\mathcal{X}^{-1} * \mathcal{X} \rightarrow \mathcal{I}; \tag{r2}$$

$$(\mathcal{X} * \mathcal{Y}) * \mathcal{Z} \rightarrow \mathcal{X} * (\mathcal{Y} * \mathcal{Z}); \tag{r3}$$

$$\mathcal{X}^{-1} * (\mathcal{X} * \mathcal{Y}) \rightarrow \mathcal{Y}; \tag{r4}$$

$$\mathcal{X} * \mathcal{I} \rightarrow \mathcal{X}; \tag{r5}$$

$$\mathcal{I}^{-1} \rightarrow \mathcal{I}; \tag{r6}$$

$$(\mathcal{X}^{-1})^{-1} \rightarrow \mathcal{X}; \tag{r7}$$

$$\mathcal{X} * \mathcal{X}^{-1} \rightarrow \mathcal{I}; \tag{r8}$$

$$\mathcal{X} * (\mathcal{X}^{-1} * \mathcal{Y}) \rightarrow \mathcal{Y}; \tag{r9}$$

$$(\mathcal{X} * \mathcal{Y})^{-1} \rightarrow \mathcal{Y}^{-1} * \mathcal{X}^{-1}. \tag{r10}$$

To deal with scalars we introduce the canonical set for abelian group given in section 7. These rules are renamed so that they have names $(r11)$ to $(r15)$. Furthermore, we introduce the six previous equations as left to right rewrite rules $(r16)$ to $(r22)$. Some new rules are then derived and the program proposes the following equation:

$$Tr(x,y,z) * (T(\theta) * Tr(u,v,w)) = Tr(x+u, y, z) * (T(\theta) * Tr(0,v,w)). \tag{E}$$

We cannot give any orientation to this rule without losing the finite termination property (take $u = 0$ in the two parts of the equality). In order to deal with this equation, we introduce the two following "abbreviations":

$$Tr(x,0,0) \rightarrow Tr_1(x);$$

$$Tr(0,y,z) \rightarrow Tr_2(y,z).$$

This is a way to deal separately with the properties of Tr on its first component and on its last two components. Using the second of these rewrite rules, (E) reduces to:

$$Tr(x, y, z) * (T(\theta) * Tr(u, v, w)) = Tr(x + u, y, z) * (T(\theta) * Tr_2(v, w)).$$

This time it is possible to give a left to right orientation to this rewrite rule without losing the finite termination property. A canonical term rewriting system consisting of 59 rewrite rules is then obtained after a long computation (20 minutes of CPU). We do not list all the intermediate rules obtained. Rules $(r1)$ to $(r15)$ are the ones given at the beginning of this section. We try to classify the other rules. First we list some properties of Tr alone:

$$Tr(x, y, z) * Tr(u, v, w) \rightarrow Tr(x + u, y + v, z + w); \tag{$r16$}$$

$$Tr(x, y, z) * (Tr(u, v, w) * \mathcal{X}) \rightarrow Tr(x + u, y + v, z + w) * \mathcal{X}; \tag{$r17$}$$

$$(Tr(x, y, z))^{-1} \rightarrow Tr((-x), (-y), (-z)). \tag{$r18$}$$

Then the definition of Tr_1 and the properties of Tr_1 alone:

$$Tr(x, 0, 0) \rightarrow Tr_1(x); \tag{$r19$}$$

$$Tr_1(0) \rightarrow I; \tag{$r20$}$$

$$Tr_1(x) * Tr_1(u) \rightarrow Tr_1(x + u); \tag{$r21$}$$

$$Tr_1(x) * (Tr_1(u) * \mathcal{X}) \rightarrow Tr_1(x + u) * \mathcal{X}; \tag{$r22$}$$

$$(Tr_1(x))^{-1} \rightarrow Tr_1((-x)). \tag{$r23$}$$

And analogously for Tr_2:

$$Tr(0, y, z) \rightarrow Tr_2(y, z); \tag{$r24$}$$

$$Tr_2(0, 0) \rightarrow I; \tag{$r25$}$$

$$Tr_2(y, z) * Tr_2(v, w) \rightarrow Tr_2(y + v, z + w); \tag{$r26$}$$

$$Tr_2(y, z) * (Tr_2(v, w) * \mathcal{X}) \rightarrow Tr_2(y + v, z + w) * \mathcal{X}; \tag{$r27$}$$

$$(Tr_2(y, z))^{-1} \rightarrow Tr_2((-y), (-z)). \tag{$r28$}$$

Then the properties of composition of Tr, Tr_1 and Tr_2:

$$Tr_1(x) * Tr_2(y,z) \rightarrow Tr(x,y,z); \tag{r29}$$
$$Tr_1(x) * (Tr_2(y,z) * \mathcal{X}) \rightarrow Tr(x,y,z) * \mathcal{X}; \tag{r30}$$
$$Tr_2(y,z) * Tr_1(x) \rightarrow Tr(x,y,z); \tag{r31}$$
$$Tr_2(y,z) * (Tr_1(x) * \mathcal{X}) \rightarrow Tr(x,y,z) * \mathcal{X}; \tag{r32}$$
$$Tr_1(x) * Tr(u,v,w) \rightarrow Tr(x+u,v,w); \tag{r33}$$
$$Tr_1(x) * (Tr(u,v,w) * \mathcal{X}) \rightarrow Tr(x+u,v,w) * \mathcal{X}; \tag{r34}$$
$$Tr(u,v,w) * Tr_1(x) \rightarrow Tr(x+u,v,w); \tag{r35}$$
$$Tr(u,v,w) * (Tr_1(x) * \mathcal{X}) \rightarrow Tr(x+u,v,w) * \mathcal{X}; \tag{r36}$$
$$Tr_2(y,z) * Tr(u,v,w) \rightarrow Tr(u,y+v,z+w); \tag{r37}$$
$$Tr_2(y,z) * (Tr(u,v,w) * \mathcal{X}) \rightarrow Tr(u,y+v,z+w) * \mathcal{X}; \tag{r38}$$
$$Tr(u,v,w) * Tr_2(y,z) \rightarrow Tr(u,y+v,z+w); \tag{r39}$$
$$Tr(u,v,w) * (Tr_2(y,z) * \mathcal{X}) \rightarrow Tr(u,y+v,z+w) * \mathcal{X}. \tag{r40}$$

The properties of $\mathcal{M}$ are then expressed by:

$$\mathcal{M} * \mathcal{M} \rightarrow \mathcal{I}; \tag{r41}$$
$$\mathcal{M} * (\mathcal{M} * \mathcal{X}) \rightarrow \mathcal{X}; \tag{r42}$$
$$\mathcal{M}^{-1} \rightarrow \mathcal{M}. \tag{r43}$$

Then the properties of $\mathcal{M}$ with the Tr's:

$$Tr(x,y,z) * \mathcal{M} \rightarrow \mathcal{M} * Tr((-x),(-y),z); \tag{r44}$$
$$Tr(x,y,z) * (\mathcal{M} * \mathcal{X}) \rightarrow \mathcal{M} * (Tr((-x),(-y),z) * \mathcal{X}); \tag{r45}$$
$$Tr_1(x) * \mathcal{M} \rightarrow \mathcal{M} * Tr_1((-x)); \tag{r46}$$
$$Tr_1(x) * (\mathcal{M} * \mathcal{X}) \rightarrow \mathcal{M} * (Tr_1((-x)) * \mathcal{X}); \tag{r47}$$
$$Tr_2(y,z) * \mathcal{M} \rightarrow \mathcal{M} * Tr_2((-y),z); \tag{r48}$$
$$Tr_2(y,z) * (\mathcal{M} * \mathcal{X}) \rightarrow \mathcal{M} * (Tr_2((-y),z) * \mathcal{X}). \tag{r49}$$

Then the properties of T alone:

$$T(0) \rightarrow \mathcal{I}; \tag{r50}$$
$$I(T(\theta)) \rightarrow T((-\theta)); \tag{r51}$$
$$T(\theta) * T(\theta') \rightarrow T(\theta+\theta'); \tag{r52}$$
$$T(\theta) * (T(\theta') * \mathcal{X}) \rightarrow T(\theta+\theta') * \mathcal{X}. \tag{r53}$$

The properties of T with $\mathcal{M}$:

$$T(\theta) * \mathcal{M} \rightarrow \mathcal{M} * T((-\theta)); \tag{r54}$$
$$T(\theta) * (\mathcal{M} * \mathcal{X}) \rightarrow \mathcal{M} * (T((-\theta)) * \mathcal{X}). \tag{r55}$$

And finally the properties of T with the Tr's:

$$T(\theta) * Tr_1(x) \to Tr_1(x) * T(\theta); \tag{r56}$$

$$T(\theta) * (Tr_1(x) * \mathcal{X}) \to Tr_1(x) * (T(\theta) * \mathcal{X}); \tag{r57}$$

$$T(\theta) * Tr(x,y,z) \to Tr_1(x) * (T(\theta) * Tr_2(y,z)); \tag{r58}$$

$$T(\theta) * (Tr(x,y,z) * \mathcal{X}) \to Tr_1(x) * (T(\theta) * (Tr_2(y,z) * \mathcal{X})). \tag{r59}$$

This is the largest example solved by our program so far.

20. Acknowledgments.

We thank D. Knuth for discovering an error in an earlier version of this paper.

21. References.

1. Ballantyne A.M. and Lankford D.S., *New Decision Algorithms for Finitely Presented Commutative Semigroups*. Report MTP-4, Department of Mathematics, Louisiana Tech. U., May 1979.

2. Degano P. and Sirovich F., *On Solving the Equivalence Problem for a Subclass of Primitive Recursive Functions*. Note Scentifiche S-79-18, Istituto di Scienze dell' Informazione, Pisa, Giugno 1979.

3. Evans T., *On Multiplicative Systems Defined by Generators and Relations 1., Normal Forms Theorems*. Proc. Cambridge Phil. Soc. 47 (1951), 637–649.

4. Huet G., *Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems*. 18th IEEE Symposium on Foundations of Computer Science (1977), 30–45. To appear, JACM.

5. Huet G. *A Canonical Proof of Correctness of the Knuth-Bendix Completion Algorithm*. Unpublished manuscript, March 1980.

6. Huet G. and Hullot J.M., *Proofs by Induction in Equational Theories with Constructors*. Unpublished manuscript, March 1980.

7. Huet G. and Hullot J.M., *Canonical Form Algorithms for Finitely Presented Algebras*. In preparation.

8. Huet G. and Oppen D., *Equations and Rewrite Rules: a Survey*. In "Formal Languages: Perspectives and Open Problems," Ed. Book R. , Academic Press, 1980. Also Technical Report CSL-111, SRI International, January 1980.

9. Hullot J.M., *Compilation de Formes Canoniques dans des Théories Equationnelles*. Forthcoming thesis, 1980.

10. Knuth D. and Bendix P., *Simple Word Problems in Universal Algebras*."Computational Problems in Abstract Algebra". Ed. Leech J. , Pergamon Press, 1970, 263–297.

11. Lankford D.S. and Ballantyne A.M., *Decision Procedures for Simple Equational Theories With Commutative Axioms: Canonical Sets of Commutative Reductions*. Report ATP-35, Departments of Mathematics and Computer Sciences, U. of Texas at Austin, March 1977.

12. Lankford D.S. and Ballantyne A.M., *Decision Procedures for Simple Equational Theories With Permutative Axioms: Canonical Sets of Permutative Reductions*. Report ATP-37, Departments of Mathematics and Computer Sciences, U. of Texas at Austin, April 1977.

13. Lankford D.S. and Ballantyne A.M., *Decision Procedures for Simple Equational Theories With Commutative-Associative Axioms: Complete Sets of Commutative-Associative Reductions*. Report ATP-39, Departments of Mathematics and Computer Sciences, U. of Texas at Austin, Aug. 1977.

14. Peterson G.E. and Stickel M.E., *Canonical Sets of Reductions for Equational Theories With Complete Unification Algorithms*. Tech. Report, Dept. of Computer Science, U. of Arizona, Tucson, Sept. 1977.